

WHAT IS CLAIMED IS:

1. A computer-readable storage medium configured to store a data structure, the data structure comprising:
 - a first lookup table having at least one entry, each of the at least one entry having an information storage portion; and
 - a second lookup table having at least one block of entries, each entry in the at least one block of entries storing next hop and prefix length information;

wherein each at least one entry in the first lookup table is indexable by a segment of an IP destination address,

the information storage portion of each of the at least one entry in the first lookup table stores next hop and prefix information when there is no route having a prefix matching the index of the entry and a prefix length greater than a predetermined value,

the data storage portion of each of the at least one entry in the first lookup table stores a value pointing to a block in the at least one block of entries in the second lookup table when there is a route having a prefix matching the index of the entry and a prefix length greater than the predetermined value, and

each entry in the block is indexable by an offset of the IP destination address.

2. The storage medium of claim 1, each of the at least one entry in the first lookup table including a marker bit indicating whether there is a route having a prefix matching the index of the entry and a prefix length greater than a predetermined value.

3. The storage medium of claim 1, wherein the storage medium is operatively connected to a configurable processor.

4. A computer-readable storage medium configured to store a data structure, the data structure comprising:

 a first lookup table having at least one entry, each of the at least one entry having a bitmap portion and an information storage portion; and

 a second lookup table having at least one entry, each entry in the at least one entry storing next hop and prefix length information;

 wherein the at least one entry in the first lookup table is indexable by a first portion of an IP destination address, and bits within the bitmap of the at least one entry are indexable by a second portion of the IP destination address,

 the information storage portion of the at least one entry stores next hop and prefix information when the total number of ones in the bitmap of the at least one entry is one of a given set of values, and

 the information storage portion of the at least one entry stores information pointing to an entry in the second lookup table when the total number of ones in the bitmap of the at least one entry is not one of the given set of values.

5. The storage medium of claim 4, wherein the first set of values includes one and two.

6. The storage medium of claim 5, wherein:

the information storage portion of the at least one entry stores one set of next hop and prefix length information when the total number of ones in the bitmap of the at least one entry is one;

the information storage portion of the at least one entry stores two sets of next hop and prefix length information when the total number of ones in the bitmap of the at least one entry is two; and

the information storage portion of the at least one entry stores information pointing to an entry in the second lookup table when the total number of ones in the bitmap of the at least one entry is more than two.

7. A method of storing information in the data structure of claim 4, the method comprising:

determining for each entry in a group of entries in a data structure, whether a value of the entry is different from a value of a previous entry,

when the entry value is different from the previous entry value, storing a first bit value in a corresponding place in a bitmap corresponding to that group of entries, and

when the entry value is the same as a previous entry value, storing a second bit value different from the first bit value in the corresponding place;

when a number of ones in the bitmap is in a first set of values, storing next hop and prefix length information in an entry storing the bitmap, the next hop and prefix length information corresponding to entries in the bitmap; and

when a number of ones in the bitmap is in a second set of values, storing an index to next hop and prefix length information corresponding to the bitmap information stored in a second data structure in the entry storing the bitmap.

8. A method of performing route lookup and packet forwarding in a communications network, the method comprising:

- receiving an incoming IP address and dividing the address into segment and offset portions;
- using a value of the segment to index to a particular entry in a first data structure;
- checking a marker bit of the entry;
- if the marker bit is zero, obtaining next hop information and prefix length information for the IP address from the remaining bits of the entry;
- if the marker bit is one,
- using the remaining bits of the entry to index to a block of entries in a second data structure,

- within the block of entries in the second data structure, using the offset to index to a particular entry, and
- obtaining next hop and prefix length information from the indexed entry in the second data structure; and

- using the next hop and prefix length information to forward a packet associated with the IP address to another location on a communications network.

9. A method of looking up a route and processing a data packet in a communications network, the method comprising:

- receiving a data packet from the communications network;
- extracting an IP address from a header of the packet;
- obtaining a segment and an offset of the IP address;
- using the segment to index an entry of the first lookup table;

computing a total number of ones in an bitmap of the indexed entry;

when the total number of ones is in a given set of values, obtaining next hop and prefix information from the information storage portion of the entry;

when the total number of ones is not in the given set of values,

using the offset to index a bit in the bitmap of the indexed entry,

determining a number of ones from the beginning of the bitmap to the indexed bit,

using the number of ones to the indexed bit to index an entry in a second lookup table,

checking a marker bit of the entry in the second lookup table,

if the marker bit is zero, obtaining next hop information and prefix length information for the IP address from the remaining bits of the entry in the second lookup table,

if the marker bit is one,

using the remaining bits of the entry in the second lookup table to index to a block of entries in a second data structure,

within the block of entries in the second data structure, using the offset to index to a particular entry, and

obtaining next hop and prefix length information from the indexed entry in the second data structure; and

using the next hop and prefix length information to forward a packet associated with the IP address to another location on a communications network.

10. The method of claim 9, wherein:

using the segment to index the entry in the first lookup table is done using a processor having an instruction which causes the processor to load an entry from the first lookup table using the segment;

computing a total number of ones in a bitmap of the indexed entry is done using an instruction of the processor which computes a total number of ones in the bitmap;

obtaining next hop and prefix information from the information storage portion of the entry is done using an instruction of the processor which extracts next hop and prefix length information from the first lookup table;

determining a number of ones from the beginning of the bitmap to the index bit is done using an instruction of the processor which computes a number of leading ones from a most significant bit to a current bit in a bitmap;

using the number of ones to the indexed bit to index an entry in a second lookup table is done using an instruction of the processor which computes the index into the second lookup table; and

obtaining next hop and prefix length information from the second lookup table is done using an instruction of the processor which gets next hop and prefix length information from the second lookup table.

11. The method of claim 9, wherein the method is performed on a configurable processor.

12. The method of claim 11, wherein the configurable processor includes custom instructions for performing the method.

13. A method of updating a data structure suitable for use in a route lookup system in a communications network, the method comprising:

- receiving an IP route having an IP address component, prefix length component and next hop component;
- checking a group of entries in a data structure indicated by the prefix length component, the group having a size determined by a length of the IP address less the prefix length; and
- performing a longest match procedure to update the group of entries to have most specific next hop and prefix length information for the group of entries;
- wherein checking includes
 - determining whether a given portion of an entry in the group of entries stores next hop and prefix information, or stores an index to a block of next hop and prefix information in another data structure; and
 - obtaining prefix length and next hop information for the entry based on the determination result.

14. A method of updating a data structure suitable for use in a route lookup system in a communications network, the method comprising:

- receiving an IP route having an IP address component, prefix length component and next hop component;
- when the prefix length component has a value less than a first predetermined value,
- determining whether an entry of an NHPL table in the data structure needs to be updated based on values of the prefix length and next hop components, and
- updating the entry of the NHPL table based on the determination;

when the prefix length component has a value greater than or equal to a second predetermined value,

identifying an entry in the data structure which matches the prefix length and next hop information,

determining whether the prefix length component in the address component is more specific than that stored in the entry, and whether the next hop component in the address component is different from a next hop component in the entry, and

updating the entry based on the determination; and

when the prefix length component is neither less than the first value nor greater than or equal to the second predetermined value,

determining entries in the data structure matching the prefix length component,

determining bits in a bitmap of the entries matching the prefix length component,

for each matching bit, determining whether the prefix length component is more specific than prefix length information associated with that bit and whether the next hop component is different from next hop information associated with that bit, and

updating an entry in the data structure corresponding to that bit based on the determination.